DW_bc_9

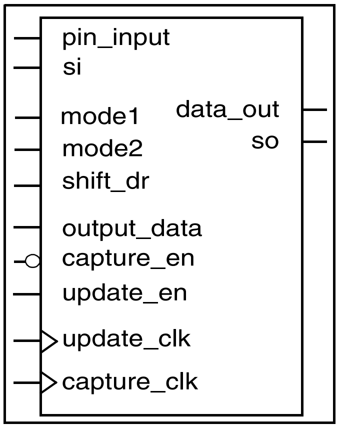
Boundary Scan Cell Type BC_9

Version, STAR, and myDesignWare Subscriptions: IP Directory

Features and Benefits

Revision History

- IEEE Standard 1149.1-2001 compliant
- Synchronous or asynchronous scan cells with respect to `tck`
- Supports the standard instructions: EXTEST, INTEST, SAMPLE/PRELOAD, and BYPASS
- Supports the optional instructions RUNBIST, CLAMP, and HIGHZ



Description

DW_bc_9 is a boundary scan cell is an output cell that observes the signal at the corresponding pin for EXTEST and observes the signal driven from the system logic for INTEST and SAMPLE. It allows a connected system network both to be driven and captured at the same pin, thus allowing such networks to be tested for shorts to others even when there are no other connected boundary scan device pins.

The Boundary Scan Description Language (BSDL) description of this cell is of type bc_9 described in the BSDL package STD_1149_1_2001.

Table 1-1 Pin Description

Pin Name	Width	Direction	Function
capture_clk	1 bit	Input	Clocks data into the capture stage
update_clk	1 bit	Input	Clocks data into the update stage
capture_en	1 bit	Input	Enable for data clocked into the capture stage, active low
update_en	1 bit	Input	Enable for data clocked into the update stage, active high
shift_dr	1 bit	Input	Enables the boundary scan chain to shift data one stage toward its serial output (tdo)
mode1	1 bit	Input	Determines whether `data_out` is controlled by the boundary scan cell or by the `data_in` signal
mode2	1 bit	Input	Determines whether `data_out` is controlled by the boundary scan cell or by the `data_in` signal
si	1 bit	Input	Serial path from the previous boundary scan cell
pin_input	1 bit	Input	IC system input pin
output_data	1 bit	Input	IC output logic signal

Table 1-1 Pin Description (Continued)

Pin Name	Width	Direction	Function
data_out	1 bit	Output	Output data
so	1 bit	Output	Serial path to the next boundary scan cell

Table 1-2 Synthesis Implementations

Implementation Name	Function	License Feature Required
str	Synthesis model	DesignWare or Test-IEEE-STD-1149-1

The DW_bc_9 cell may be synchronous or asynchronous with respect to `tck` (Test Clock system pin), depending on the port connections. Table 1-4 on page 3 lists the connections for asynchronous boundary scan chains. Table 1-5 on page 3 lists the connections for synchronous boundary scan chains.

The `mode1 and mode2` signal gives the Test Access Port (TAP) instructions control of the boundary scan cell. Table 1-3 on page 2 lists the required values of the `mode1 and mode2` signal for each of the TAP instructions that DW_bc_9 supports. The INTEST instruction is not supported if the cell is used as an output cell.

Table 1-3 Mode Signal Generation for DW_bc_9

Instruction	Mode1	Mode2
EXTEST	1	1
SAMPLE	X	0
PRELOAD	0	0
CLAMP	X	1
INTEST	0	1
RUNBIST	X	1
BYPASS	0	0

Table 1-4 lists the connections for asynchronous boundary scan chains.

Table 1-4 Port Connections for Asynchronous Boundary Scan Chains

DW_bc_9 Port Name	Connection
capture_clk	clock_dr from TAP controller
update_clk	update_dr from TAP controller
capture_en	Logic zero
update_en	Logic one
shift_dr	shift_dr from TAP controller
mode1	Mode1 generation logic
mode2	Mode2 generation logic
si	so from previous boundary scan cell
pin_input	System input pin for input cells or IC output logic for output cells
output_data	IC output logic
data_out	System output pin
so	si of next boundary scan cell

Table 1-5 lists the connections for synchronous boundary scan chains.

Table 1-5 Port Connections for Synchronous Boundary Scan Chains

DW_bc_9 Port Name	Connection
capture_clk	tck from system pin
update_clk	tck_n from system pin
capture_en	sync_capture_en from TAP controller
update_en	sync_update_dr from TAP controller
shift_dr	shift_dr from TAP controller
mode1	Mode1 generation logic
mode2	Mode2 generation logic
si	so from previous boundary scan cell
pin_input	System input pin for input cells or IC output logic for output cells

Table 1-5 Port Connections for Synchronous Boundary Scan Chains (Continued)

DW_bc_9 Port Name	Connection
output_data	IC output logic
data_out	System output pin
so	si of next boundary scan cell

Suppressing Warning Messages During Verilog Simulation

The Verilog simulation model includes macros that allow you to suppress warning messages during simulation.

To suppress all warning messages for all DWBB components, define the DW_SUPPRESS_WARN macro in either of the following ways:

- Specify the Verilog preprocessing macro in Verilog code:

  ```
  `define DW_SUPPRESS_WARN
  ```

- Or, include a command line option to the simulator, such as:

  `+define+DW_SUPPRESS_WARN` (which is used for the Synopsys VCS simulator)

The warning messages for this model include the following:

- If values other than 1 or 0 are present on a clock port, the following message is displayed:

  ```
  WARNING: <instance_path>.<clock_name>_monitor:
      at time = <timestamp>, Detected unknown value, x, on <clock_name> input.
  ```

  To suppress only this warning message for all DWBB components, use the following macro:

  - ❑ Define the DW_DISABLE_CLK_MONITOR macro. You can define this macro in the following ways:
    - Specify the Verilog preprocessing macro in Verilog code:

      ```
      `define DW_DISABLE_CLK_MONITOR
      ```

    - Or, include a command line option to the simulator, such as:

      `+define+DW_DISABLE_CLK_MONITOR` (which is used for the Synopsys VCS simulator)

  This message is also suppressed using the DW_SUPPRESS_WARN macro explained earlier.

Related Topics

- Application Specific – JTAG Overview
- DesignWare Building Block IP User Guide

HDL Usage Through Component Instantiation - VHDL

```
library IEEE,DWARE;
use IEEE.std_logic_1164.all;
use DWARE.DW_Foundation_comp_arith.all;

entity DW_bc_9_inst is
  port (inst_capture_clk : in  std_logic;
        inst_update_clk  : in  std_logic;
        inst_capture_en  : in  std_logic;
        inst_update_en   : in  std_logic;
        inst_shift_dr    : in  std_logic;
        inst_mode1       : in  std_logic;
        inst_mode2       : in  std_logic;
        inst_si          : in  std_logic;
        inst_pin_input   : in  std_logic;
        inst_output_data : in  std_logic;
        data_out_inst    : out std_logic;
        so_inst          : out std_logic  );
end DW_bc_9_inst;

architecture inst of DW_bc_9_inst is
begin
  -- Instance of DW_bc_9
  U1 : DW_bc_9
    port map (capture_clk => inst_capture_clk,
              update_clk  => inst_update_clk,
              capture_en  => inst_capture_en,
              update_en   => inst_update_en,
              shift_dr    => inst_shift_dr,
              mode1       => inst_mode1,
              mode2       => inst_mode2,
              si          => inst_si,
              pin_input   => inst_pin_input,
              output_data => inst_output_data,
              data_out    => data_out_inst,
              so          => so_inst   );
end inst;

-- pragma translate_off
configuration DW_bc_9_inst_cfg_inst of DW_bc_9_inst is
  for inst
  end for; -- inst
end DW_bc_9_inst_cfg_inst;
-- pragma translate_on
```

HDL Usage Through Component Instantiation - Verilog

```
module DW_bc_9_inst(inst_capture_clk, inst_update_clk, inst_capture_en,
                    inst_update_en, inst_shift_dr, inst_mode1, inst_mode2,
                    inst_si, inst_pin_input, inst_output_data, data_out_inst,
                    so_inst );
  input inst_capture_clk;
  input inst_update_clk;
  input inst_capture_en;
  input inst_update_en;
  input inst_shift_dr;
  input inst_mode1;
  input inst_mode2;
  input inst_si;
  input inst_pin_input;
  input inst_output_data;
  output data_out_inst;
  output so_inst;

  // Instance of DW_bc_9
  DW_bc_9
    U1 (.capture_clk(inst_capture_clk),
        .update_clk(inst_update_clk),
        .capture_en(inst_capture_en),
        .update_en(inst_update_en),
        .shift_dr(inst_shift_dr),
        .mode1(inst_mode1),
        .mode2(inst_mode2),
        .si(inst_si),
        .pin_input(inst_pin_input),
        .output_data(inst_output_data),
        .data_out(data_out_inst),
        .so(so_inst) );
endmodule
```

Revision History

For notes about this release, see the *DesignWare Building Block IP Release Notes*.

For lists of both known and fixed issues for this component, refer to the STAR report.

For a version of this datasheet with visible change bars, click here.

Date	Release	Updates
July 2020	DWBB_201912.5	■ Adjusted content and title of "Suppressing Warning Messages During Verilog Simulation" on page 4 and added the DW_SUPPRESS_WARN macro
October 2019	DWBB_201903.5	■ Added the "Disabling Clock Monitor Messages" section ■ Added this Revision History table and the document links on this page

Copyright Notice and Proprietary Information

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at https://www.synopsys.com/company/legal/trademarks-brands.html.

All other product or company names may be trademarks of their respective owners.

Free and Open-Source Software Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.
www.synopsys.com